“Java™: Fad or Future”

Karl Mueller/520

David Fout/Century Computing Inc.

June 5, 1997

Agenda

- What Exactly is Java?
- Java History
- Applets vs. Applications
- Object-Oriented Concepts
- Java Language
- Future Initiatives
- Advantages/Issues
- Applicability at GSFC

What Exactly is Java?

- **Object-oriented programming language**
  - **Borrows from C, C++, Objective C, Ada, SmallTalk, Common Lisp**
- **Simple Programming Paradigm**
- **Platform Independent**
  - **Write Once, Run Anywhere™**
  - **Virtual Machine**
- **Distributed**
- **Robust**
- **Built-in Security**
- **Multithreaded**
- **Dynamic**

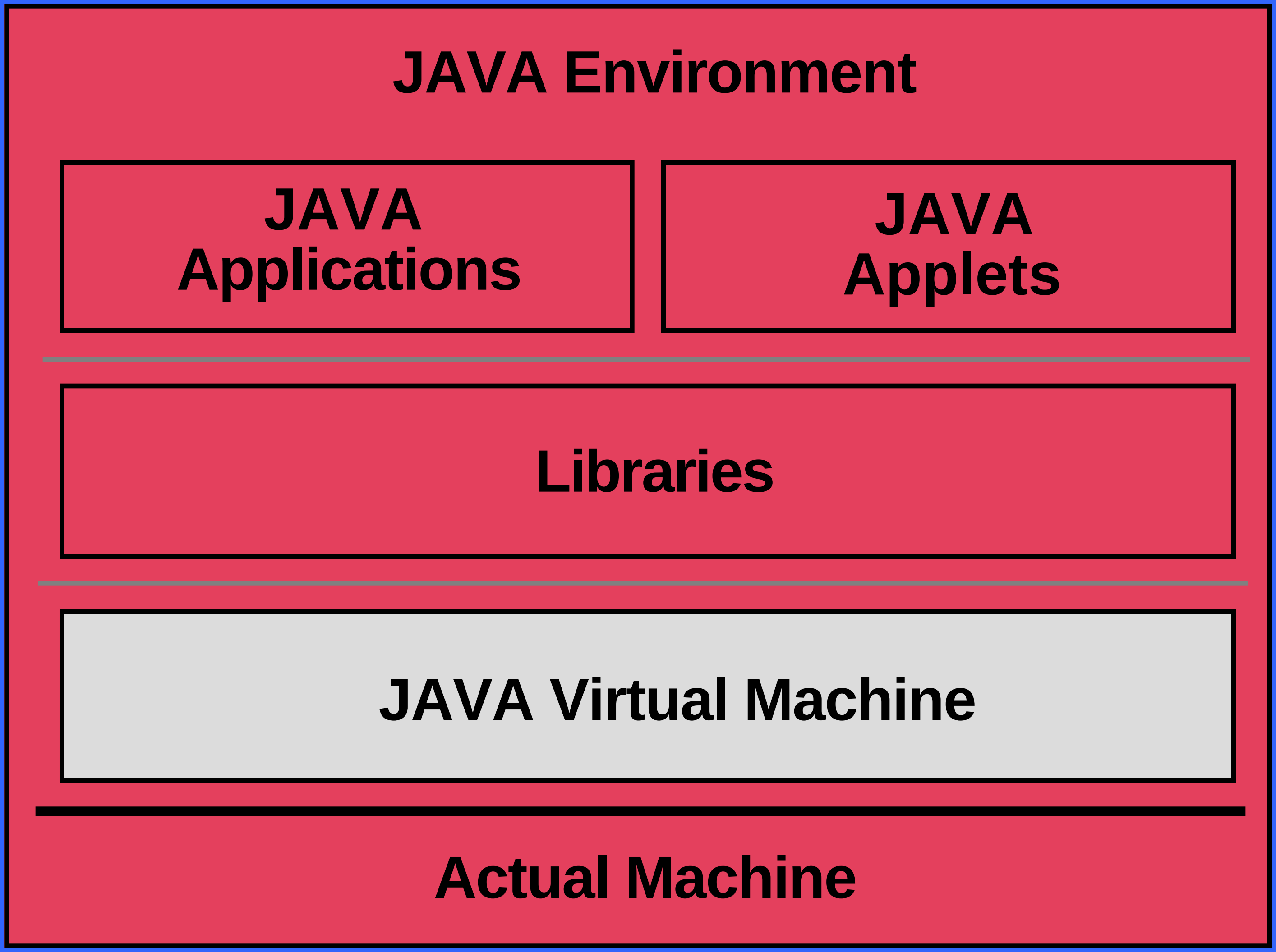
Java History

- Requirement for a Small Computer Language (1991)
  - Programs would run on consumer devices (e.g., cable TV switchboxes)
- Small Memory Usage Requirements
  - Tight code
- Program needed to be processor-independent
- Resurrected the UCSD Pascal virtual machine model
  - Pioneered by Niklaus Wirth
- James Gosling led a team (called“Green” team) to create “Oak”
  - Based on C++ and object-oriented (1992)
  - Renamed to “Java” because “Oak” was already a language

Java History (Cont’d)

- In 1992 the “Green” Project Produced “*7”
  - Intelligent remote control
  - Vigorously tried to market their product with no success
- In 1994 the World Wide Web became a phenomenon
- James Gosling and his team decided to build a “real cool browser”
  - Architecturally neutral, real-time, secure
  - HotJava browser written entirely in Java and interpreted Java bytecode (Java Applets)
  - “Proof of Technology” shown at SunWorld ‘95
- Breakthrough for Java came with the release of Java-enabled Netscape (Netscape 2.0)
  - IBM, Symantec, Borland, and even Microsoft license Java technology

Java Virtual Machine



JAVA Environment

JAVA Applications

JAVA Applets

Libraries

JAVA Virtual Machine

Actual Machine

Java Libraries (Packages)

- Java Packages
  - java.applet
  - java.awt
  - java.beans (New)
  - java.io
  - java.lang
  - java.lang.reflect (New)
  - java.math (New)
  - java.net
  - java.rmi (New)
  - java.security (New)
  - java.sql (New)
  - java.text (New)
  - java.util
  - java.util.zip (New)

Applets vs. Application

Applets vs. Applications

- Applet
  - Runs within the context of a web browser
  - Must be defined as a subclass of the class “Applet”
  - Security “Sandbox model”
- Application
  - Runs “stand alone” like a typical non-Java application (e.g., Microsoft Word)
  - Requires the “main” method in the controlling class

Sample Java Application

```
/**
* The HelloPlanet class implements an application that simply
* displays “Hello Beautiful Blue Planet!” to the standard
   output.
*/
class HelloPlanet
{
   public static void main(String[] arguments)
   {
       System.out.println(“Hello Beautiful Blue Planet!”);
   }
}
```

Java Applets

- Can be downloaded from the World Wide Web (WWW) and run in any Java-enabled WWW browser.
- Configured using HTML.
- Give dynamic content to web pages.
- Must implement at least one of the following methods:
  - init
  - start
  - paint

Sample Applet
HTML Page

```
<HTML>

<TITLE>Hello Planet Sample Applet</
  TITLE>

<BODY>

<APPLET CODE=“HelloPlanet.class” WIDTH
  = 200 HEIGHT=200>

</APPLET>

</BODY>

</HTML>
```

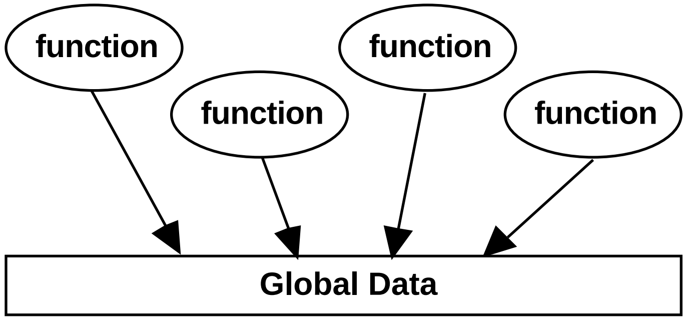
Sample Java Applet

```
import java.awt.*;
import java.applet.*;
public class HelloPlanet extends Applet
{
   public void paint(Graphics graphics)
   {
       Font font = new Font(“System”, Font.BOLD, 18);
       graphics.setFont(font);
       graphics.drawString(“Hello Beautiful Blue Planet.”,
                    25, 50);
   }
}
```

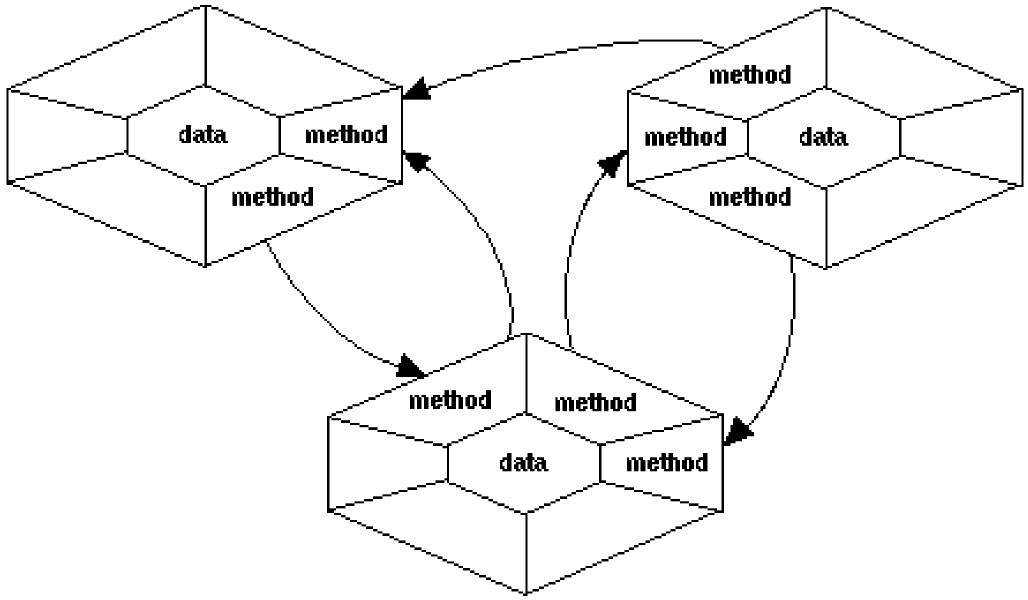
Object-Oriented Concepts

Procedural vs. Object Oriented

Procedural Programming



Object-Oriented Programming



Object-Oriented Concepts

- Classes
- Objects Contain:
  - Data:
    - Define object’s state
  - Methods:
    - Define object’s behavior
- Class Relationships:
  - Use
  - Contain (“has-a”)
  - Inheritance (“is-a”)
- Object-Oriented Languages Must Have:
  - Encapsulation (i.e., information hiding)
  - Inheritance
  - Polymorphism (i.e., run-time binding)

Java Language

Java Language

- Data Types:
  - Java is strongly typed.
  - Primitive Types are not Objects.
- Variables
- Assignments and Intializations
- Constants
- Operators:
  - Arithmetic
  - Relational & Boolean
  - Bitwise

Java Language (Cont’d)

- Control Flow:
  - Conditional Statements
  - Indeterminate Loops
  - Determinate Loops
  - Switch Statement
  - Labeled/Unlabeled Breaks

Sample Java Code

```
int exponentValue = fMaximumDepth;
for (int i = 0; i <= fMaximumDepth; i++)
{
   int leadingNumberOfSpaces = (int) Math.pow((double) 2,
              (double) exponentValue) - 1;

   int j = 0;
   while (j < leadingNumberOfSpaces)
   {
       System.out.print(" ");
       j++;
   }

   System.out.println();

   exponentValue--;
}
```

Class Methods & Variables

```
public class UserPreferences
{
   public UserPreferences()
   {
        fColor = Color.GREEN;
   }

   public UserPreferences(int Color)
   {
        fColor = Color;
   }

   public int getColor() { return fColor; }
   public setColor(int Color) { fColor = Color; }

   private int fColor;
}

// Create an instance of UserPreferences
int myColor = Color.VIOLET;
UserPreferences myPreference = new UserPreferences(myColor);
```

Input/Output (Streams)

- **InputStream Can Read a Sequence of Bytes.**
- **OutputStream Can Write a Sequence of Bytes.**
- **Many Stream Types are Derived from InputStream or OutputStream:**
    - **Example: FileInput/Output, DataInput/Output, ObjectInput/Output, etc.**
- **Object Serialization:**
    - **Convert an object to a stream of bytes to be read or written.**
    - **Persistence implemented by ObjectInput/Output stream.**
    - **Same paradigm to save to a file or across a network.**

Exceptions

- Mechanism to Accommodate Errors/ Exceptional Occurrences.
- All Exceptions Throw an Instance of Class Derived from Throwable:
  - Derived Error Class:
    - Internal Java Errors.
  - Derived Exception Class:
    - Some type of abnormal program error.
- Method Must Declare All Explicit Exceptions it Throws.
- Exceptions May Be Caught Anywhere Up the Call Tree.
- Code In the “Finally” Clause is Always Executed.

MultiThreading

- Multithreading is Built into Java
- Create New Thread of Execution by Extending the “Thread” Class
  - Place code in the “Run” method
  - Calling method Invokes the thread by calling the “Start” method
- Can Start and Stop a Thread
- Thread Priorities
- Synchronization
  - Can cause deadlocks

Networking

- Establish Network Connection by Opening a Socket (TCP, UDP):
  - Client Socket
  - Server Socket
- Reading from a File and a Socket Uses a Similar Paradigm.
- Java has Built-in Support to Connect To Uniform Resource Locators (URL).

Database Connectivity (JDBC)

- JavaSoft Provided Pure Java API for SQL:
  - Also provided device manager to allow third party drivers to connect to a particular database.
- Normal Security Restrictions for Applets Apply with Database Access.
- Based on X/Open SQL Call Level Specification.
- Easily Move Data Between Different Database Vendors.

Remote Method Invocation (RMI)

- Invoke Methods Associated with an Object Located on a Different Machine:
  - Java-to-Java communication.
  - Whole process completely transparent to the Java programmer.
- RMI Utilizes the Object Serialization Mechanism.

Native Methods

- Call Other Languages from Java.
- Call Java Methods from Other Languages.
- Reasons to Use Native Methods:
  - Substantial Amounts of Legacy Code Exist.
  - Application requires access to system features which are difficult or impossible to implement in Java.
  - Need to maximize the execution speed.
- Lose Portability:
  - Write Once, Run Anywhere no longer applies.
  - Must supply separate method library for every supported platform.
  - Easy to Corrupt Java Virtual Machine or Worse.

Abstract Windowing Toolkit (AWT)

- The AWT provides the user interface for Java programs.
- Peers:
  - Java program always has the look and feel of the platform that you are on.
- Components
- Containers
- Layouts
- Events:
  - Old model (JDK 1.0)
  - New model (JDK 1.1)

New Features of AWT

- Improved Event Handling
- New Components and Containers:
  - Java 1.1 provides one new component, the Popup Menu, and one new Container, the ScrollPane.
- Menu Shorts
- Clipboards
- Printing
- Access to Color Schemes
- Graphics
- New Method Names
- components->JavaBeans:
  - The @deprecated tag

JavaBeans Component Architecture

- "Write once, re-use everywhere!"
- JavaBeans is the platform-neutral, component architecture for Java.
- It provides the capability to build reusable components by defining standards for the way objects talk to each other.
- Individual Beans will function quite differently, but typical unifying features that distinguish a Bean are:
  - Introspection
  - Customization
  - Events
  - Properties
  - Persistence
  - Packaging

Internet Development Environments (IDE)

- Most IDEs that support JDK 1.1 will support JavaBeans. They also provide their own industrial strength JavaBeans components.
- The following is a list of Beans Compliant Builders:
  - Asymetrix's SuperCede
  - Borland's JBuilder
  - Cosmo Suite (Code)
  - IBM's AppletAuthor
  - IBM's Visual Age for Java
  - Lighthouse Design's JavaPlan
  - Netscape's "Palomar"
  - Penumbra's Mojo
  - SunSoft's Project Studio
  - SunSoft's Java Workshop
  - Sybase's "Jato" (formerly "Starbuck")
  - Symantec's Visual Cafe
  - Tek-Tools' Kawa
  - Unify's Vision

See http://java.sun.com/beans/tools.html

Java Security API

- The Java Security API is a new Java core API, built around the java.security package.:
  - Digital Signatures (signed applets) -digital signature algorithms, such as DSA. The functionality includes generating public/private key pairs as well as signing and verifying arbitrary digital data.
  - Message Digests - producing "digital fingerprints" of data; provide unique and non-forgeable identifiers of digital data.
  - Key Management -allows applications to design their own key management systems, and to interoperate with other systems at a high level.
  - Access Control Lists (ACL) - a data structure that guards access to resources.
- JDK 1.1 provides a tool that can sign Java Archive (JAR) files, which can contain classes and other data (such as images and sounds). Using the digital signatures you can define trusted applets, which allows any downloaded applets in JAR files signed (using the tool) by a trusted entity to run with the same full rights as local applications. The applets will not be subject to the "sandbox" restrictions of the original Java security model.

Java Native Interface (JNI)

- The Java Native Interface provides a consistent interface between Java code and native (platform-specific) code. It replaces the Native Method Interface in JDK 1.0. JNI has two main uses:
    1. It specifies a way to write Java native methods.
    2. It includes an invocation API embedding a JVM in native applications.
- Previously, Java used Microsoft’s Native Method Interface.
- Netscape plans to support JNI in the Communicator 4 release.

Future Initiatives

JavaBeans

- Beans Development Kit (BDK):
  - The Beans Development Kit was released in February 97. It includes:
    - BeanBox sample Bean container - (This can be used to "test" JavaBeans that you develop.):
      - Example JavaBeans
      - Sample source code
      - JavaBeans documentation
      - JavaBeans tutorial
  - JavaBeans Bridge for ActiveX:
    - A JavaBeans Bridge for ActiveX Bridge is available in Beta form at the SunSoft Web site. Currently, it works in Word, Excel, Internet Explorer and Visual Basic. The final version will be bundled in the BDK.
    - JavaBeans Migration Assistant for ActiveX.
    - Sun will also be adopting IBM's JavaBeans Migration Assistant for ActiveX, a tool that converts ActiveX controls and into JavaBeans (wrappers). This tool analyzes an ActiveX control's properties and creates a 100% Pure Java container that takes on the features of the ActiveX control and then implements the component functions. JavaSoft will include the JavaBeans. A beta is expected in May and a final version is expected in July.

Java Runtime Environment (JRE)

- The Java Runtime Environment consists of the Java Virtual Machine, Java Core Classes and supporting files.
- It contains no compiler, no debugger, and no tools.
- This product satisfies the demands for the smallest runtime to be bundled with Java applications.

Java Media Framework API (JMF)

- Java Sound
- Java Speech
- Java Advanced Imaging
- Java Animation
- Java Collaboration
- Java Media Player
- Java Telephony API (JTAPI)
- Java 2D
- Java 3D

3D on the Web Today

- Java 3D:
  - Looks to be extremely promising but will not be available for at least another year.
  - If you wish to do dynamic 3D on the web today, you must look elsewhere. Fortunately, other contemporary technologies are available.

3D on the Web Today (Cont'd)

- VRML:
  - The Virtual Reality Modeling Language (VRML) is a file format for describing 3D interactive worlds and objects.
  - VRML is capable of representing static and animated objects and can have hyperlinks to other media such as sound, movies, and images.
  - Interpreters (browsers) for VRML are widely available for many different platforms as well as authoring tools for the creation VRML files.
  - While VRML 1.0 was primarily static, VRML 2.0 provides the ability to build "Moving Worlds". It offers engines, timers, collision detection and script nodes that allow for a truly dynamic scene.
  - VRML97 will be the next release of the VRML specification. It is almost identical to VRML 2.0, but with numerous improvements to the specification document and a few minor functional differences.

3D on the Web Today (Cont’d)

- Script Nodes:
  - Using Script nodes in VRML 2.0, you can use Java classes to affect the VRML world. With Script nodes, you are essentially creating a new node with associated fields and events. Those fields and events can be passed to a specified Java class. That class then has access to manipulate any node passed to it from the VRML script node.
  - Current VRML browsers that support Java and VRML 2.0:
    - Silicon Graphics CosmoPlayer
    - Dimension X's Liquid Reality
    - Sony's Community Place (formerly known as CyberPassage).
  - Script nodes offer limited functionality because each Script node creates a new instance of a Java applet. There is no way to share the same Java applet among Script nodes or have the Script node "attach" to a control applet. For this reason, Script nodes are not an efficient way to connect data to VRML scenes. They are good for adding logic and physics to a scene.

3D on the Web Today (Cont'd)

- **External Authoring Interface (EAI):**
  - **EAI is a specification that provides a method for developing applications that interact with, and dynamically update a 3D scene.**
  - **These outside applications can "reach" into the VRML scene and manipulate it. The EAI is in the process of adoption into the VRML 2.0 specification.**
  - **SGI's CosmoPlayer is the only current implementation.**
  - **Currently, it is considered an "Informative Annex"; meaning browsers may include the EAI into their functionality and remain compliant to the VRML specification.**

3D on the Web Today (Cont’d)

- Liquid Reality:
  - Dimension X's Liquid Reality SDK will include the Liquid Reality platform for Java and VRML development on the Internet.
  - Liquid Reality provides a development environment of over 450 Java classes that developers can leverage or extend to create 3D applications and content.
  - It also provides developers the tools to create custom 3D applications for browsing, manipulating and creating VRML 2.0 content.

AWT

n New Components: Java Foundation Classes (JFC):

– Licensed Netscape’s Internet Foundation classes:

- ❖ tree view
- ❖ tab folders
- ❖ toolbars
- ❖ pane splitters
- ❖ color pickers
- ❖ font choosers
- ❖ icons
- ❖ status bars
- ❖ message boxes
- ❖ tool tips (online help)
- ❖ sliders
- ❖ gauges
- ❖ spinners

n Drag and Drop:

– Drag and Drop will work on top of the new Clipboard API. It will provide the capabilities to drag and drop to and from Java and native applications.

AWT: Pluggable Look and Feel

- Peerless components:
  - Currently, a components peer defines the look of a component.
- The developer will be able to specify the look and feel for an entire application, a specific component type, or a single component and offer the ability for the user to select from a variety of look and feels.
- JDK will provide look and feels for major user interfaces (Windows 95, Mac, and Motif).
- You will also have the ability to build your own complete sets of look and feels.
- Vendors can publish look and feels and you will be able to download them (NextStep, etc...).

Security

- Fine granularity of security:
  - By expanding the access control lists, users can assign very specific permissions to individual applications or code signers, such as the ability to read or write only within specific directories or to connect to particular ports on certain hosts.

The Java Server API

- Java Web Server (Jeeves):
  - First 100%Pure Java web server
  - Java enabled
- The Java Server API:
  - Toolkit to develop a wide variety of network aware servers
  - FTP, http, etc...
- Java Servlet API:
  - Server side extensions
  - Replacement for cgi-bin

JAVAOS

- “Small and efficient”.
- JavaOS provides a runtime specifically tuned to run Java applications directly on hardware platforms without requiring a host operating system.
- The JavaOS architecture consists of a microkernel and memory manager, device drivers, the Java Virtual Machine, the JavaOS Graphics and JavaOS Windowing systems, networking classes and support for the full Java API.
- Applications written for JavaOS can also run using Java-enabled on any platform that contains the Java VM.

HotJava Views

- HotJava Views a highly integrated, easy-to-use WebTop environment:
  - E-mail
  - Calendar
  - Web access
- It allows companies to easily integrate in-house or third party Java solutions seamlessly into the user environment.
- OEMs will be able to incorporate HotJava Views in the products they ship.
- HotJava Views is JavaSoft's first user environment that runs on top of JavaOS.
- HotJava Views will run on any server that contains the Java Virtual Machine.

Java Embedded API

- Embedded Java will be a version of the Java virtual machine designed for embedded implementations in devices such as pagers, routers, and switches.
- It will have very low RAM requirements, and will require only .5 MB of ROM.
- A reference platform for Embedded Java is to arrive in the fourth quarter of 1997.
- The Java Embedded APIs specify how the Java API may be subsetted for embedded devices that are incapable of supporting the full Java Core API:
  - It includes a minimal embedded API based on java.lang, java.util and parts of java.io.
  - It then defines an additional series of extensions for particular areas such as networking and GUIs.

Java Commerce API

- Java Commerce API will bring secure purchasing and financial management to the Web.
- JavaWallet is the initial component, which defines and implements a client-side framework for credit card, debit card, and electronic cash transactions.
- The following is list of JECF components:
  - Infrastructure - provides the interaction mechanism between the other components.
  - Database - stores permanent records of personal information, transaction history, and a list of the types of payment instruments the user has available.
  - Payment/Service Cassettes - are implementations of the various payment instruments and their associated protocols.
  - Administrative Interfaces - the dialog windows and configuration units that control and customize the behavior of your JECF system as a whole.
- The core functions of user identification, authentication, and authorization are performed locally on your machine. Password protection and unique user information is performed and stored on your system, and none of this data is sent over the network.

Just In Time Compilers (JIT)

- Accelerates execution performance up to 50 times.
- Compiles byte-code to machine code.
- Built into the Virtual Machine (transparent).
- Really pays off when same code is re-executed.

Push Technology

- Push technology reversed the traditional Web-based paradigm by delivering or "pushing" the information to your computer automatically.
- This technology is being billed as the second wave of the World Wide Web.
- It attempts to solve two fundamental problems:
  - Access to too much information:
    - Ex. Stock ticker application
  - The expensive and slow process of software deployment and installation:
    - Ex. Bug fixes, new releases
- Off-line browsing

Advantages/Issues

Java Advantages

- Object Oriented (reuse, modularity,maintainability)
- Built-in garbage collection / improved memory management
- Strong typing
- Threading is part of the language
- Object persistence
- Reflection
- Java Security Model
- Large library of classes provided with the language
- Platform independent (almost)
- Built-in network programming
- Single point of control for language specification
- Substantial industry investment and support

Java Issues

- Some features not completely consistent across platforms:
  - AWT
  - Threads
- Interpreted - performance not yet proven.
- JDK 1.1 Supports:
  - Browsers
  - IDE’s
- Risk of new technology.
- One company (Sun) controls the standard - direction unclear.
- Not able to live up to high expectations.
- Rapidly changing.
- Microsoft competition.

Applicability at GSFC

For Today's New Projects

- So, if you're starting development this FY on a new system, should you consider Java?

  YES!

- Some deciding questions are:
  - How critical are your performance requirements?
    - *Java may not be ready for some real-time applications*
  - What operating system are you running?
    - *Does your OS have a JVM*
  - Do you need specific libraries existing in another language?
  - Can you obtain staffing with the necessary expertise?
    - *Experienced OO programmers (i.e., C++) can easily pick up Java. Programmers new to OO will pick up Java more easily than C++, but there will be a learning curve to adopt the new OO paradigm.*
  - Can you assume and mitigate a minimal amount of risk due to the evolving nature of the technology?
- *Java is not limited to Web browser-based UI's*

Java: Fad or Future?

The Code 520 ‘experts’ say

FUTURE

Here are some of their comments:

– “Java will be a big part of the computing industry’s future until something supersedes its core capabilities. Maybe an object/agent-oriented language.”

– “Java is not ready for real-time or low-level tasks. Neither is it ready for tasks that require great speed. Data interaction, data distribution, data analysis, and scheduling (to name a few) can be implemented in Java.”

– “Java will dominate the next 10 years (estimate 3 years to catch c++ in number of developers)” as the new development language of choice.

– “Java is going to be around for some time. Too many people have a lot of money tied to it. It is the best language today to write distributed, platform independent applications.”

– “Java will probably replace C++ in a few years. I can’t think of any reason why it wouldn’t (unless Microsoft succeeds in burying it).”

Code 522 Java Style Guide

- C++ Style Guide used on existing projects:
  - GenSAA, TPCE, DSDP
- Future projects will use Java instead of C++.
- Needed new Style Guide for Java.
- Based on the C++ Style Guide.
- Goals: Enhance reliability and maintainability of Java code.

Code 522 Java Style Guide (Cont’d)

- Covers broad range of issues:
  - Consistent formatting:
    - White space, comments, naming conventions.
  - File organization:
    - Packages, prologs, order of contents.
  - Use of language constructs:
    - Import statements, methods, variables, threads.
  - Tips and techniques:
    - General Java hints and suggestions.